LLM alignment fine-funining: DPO vs IPO

Tri Nguyen

nguyetr9@oregonstate.edu

March 27, 2025

1 "Notations"

- a LLM **generates** responses: perform certain decoding techniques given a language model. Popular choices: beam search, top-k sampling, top-p sample.
-

2 What is LLM fine-tuning and why

Large language models (LLMs) are becoming significantly popular and bring broad impacts across various aspects of our life. One of the key factors attributed to this success is the capability of a model to perform on very wide range of tasks: you can just input a task description and the relevant information and feed it to the model as a prompt. This ability is commonly referred to LLM being able to acquire various skills, or being able to follow instruction, or can be seen as LLM's generalization power careful here.

This amazing performance can be largely attributed to the fine-tuning steps, where the pretrained model is twisted to learn to produce outputs that are more aligned with human (or the creator's goals), whether it is to follow instruction, to produce "better response", including being helpful, harmless, factual, or refuse to response to malicious requests. The very first work that shakes the whole NLP/AI community is from OpenAI, where they kind of lay out the roadmap to do fine-tuning:

1. Supervised fine-tuning. Assume access to demonstration dataset: $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^N$ where $\boldsymbol{y}_n$ is the ideal response given the prompt $\boldsymbol{x}$. Example here pls. The objective in the step is to twist the model to produce more response like in the datasets. This is supervised learning objective and can be accomplished as any next-token predicting task.

   The challenge in this step is not about the method but about the dataset. It is considerably hard to collect the ideal response that is varying over the whole prompt space. Even OpenAI can only collect [[**x**]] pair of samples to do this. Therefore the need of second step, which is the focus of this article.

2. Preference learning. There are some variants, but many works considers a using pairwise preference dataset: $\left\{\boldsymbol{x}_n, \boldsymbol{y}_n^{(1)}, \boldsymbol{y}_n^{(2)}, c_n\right\}$, where given a prompt $\boldsymbol{x}_n$, and 2 possible responses $\boldsymbol{y}_n^{(1)}, \boldsymbol{y}_n^{(2)}$, the preference label $c_n \in \{1, 2\}$ indicates which response is preferred over the others.

   Example: This type of dataset is arguably easier to collect: one can use pretrained LLMs to generate 2 different responses given the prompt $\boldsymbol{x}$, then ask a human annotator to choose which one is better. The particular details such as where to collect $\boldsymbol{x}$, how to define preference are depending on specific tasks. And although many works tend to refer the problem as human preference fine-tuning, preference can be anything, not necessarily based on human.

   Using this dataset, the popular work Reinforcement Learning with Human Feedback (RLHF) proposed to formulate the problem of fine-tuning LLM as:

$$\underset{\pi}{\text{maximize}} \quad \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}}\left[\mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi}\left[s(\boldsymbol{x},\boldsymbol{y}) - \beta D_{\mathsf{kl}}(\pi \parallel \pi_{\text{ref}})\right],\right] \tag{1}$$

where the *score function* $s(\boldsymbol{x},\boldsymbol{y})$ outputs a scalar value indicating how strongly aligned the response $\boldsymbol{y}$ is given prompt $\boldsymbol{x}$. The objective in English is: learn a policy (a language model) π that behaves not so much differently from π_{ref} and π generates responses $\boldsymbol{y}$ with highest possible $s(\boldsymbol{x},\boldsymbol{y})$. π_{ref} is a reference policy which can be seen as a good initialization to start with. In practice, π_{ref} is the trained policy obtained from step 1.

Having the overall view of the fine-tuning pipeline, we are ready to dive what the existing works proposed.

3 RLHF and DPO

3.1 RLHF

While the title didn't mention RLHF, we start from RLHF to see the cleverness in the development of DPO. The very first obstacle in optimizing (1) is that score function s is **unknown**. RLHF proposed to solve this as an independent problem. If we assume the preference label follow the Bradley-Terry (BT) model, we have

$$\mathsf{Pr}\left(c = 1 \mid \boldsymbol{x}, \boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}\right) = \frac{\exp(s(\boldsymbol{x},\boldsymbol{y}^{(1)}))}{\exp(s(\boldsymbol{x},\boldsymbol{y}^{(1)})) + \exp(s(\boldsymbol{x},\boldsymbol{y}^{(2)}))} = \sigma(s(\boldsymbol{x},\boldsymbol{y}^{(1)}) - s(\boldsymbol{x},\boldsymbol{y}^{(2)})) \tag{2}$$

As the score function gives a higher score for better aligned response, it is pretty reasonable to employ the BT model.

Pairwise Generative Model. Under this model, the problem of learning s can be cast as a somewhat special binary classification problem: Given a sample $(\boldsymbol{x},\boldsymbol{y}^1,\boldsymbol{y}^2)$, we wish to learn $h_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}^1,\boldsymbol{y}^2)$ to predict the true binary label c.

$$\mathsf{Pr}\left(c = 1 \mid \boldsymbol{y}^a, \boldsymbol{y}^b, \boldsymbol{x}\right) = \sigma(s^\natural(\boldsymbol{y}^a,\boldsymbol{x}) - s^\natural(\boldsymbol{y}^b,\boldsymbol{x})) \tag{3}$$

There are some specific details that a general classification problem does not apply here:

1. Firstly, because of the underlying model given in (2), $h_{\boldsymbol{\theta}}$ need to be parameterized as

   $$h_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}^1,\boldsymbol{y}^2) \triangleq \sigma\left(s_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}^1) - s_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}^2)\right)$$

   where $s_{\boldsymbol{\theta}}$ can be parameterized using a LLM. The argument for the use of LLM is that LLM is so large, it has the potential to work on any task, including giving a score to a pair of prompt, response. Then maximum likelihood estimation gives us

   $$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad -\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}_1,\boldsymbol{y}_2}\left[\mathbb{I}[c=1]\log\sigma(h_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}^1,\boldsymbol{y}^2)) + \mathbb{I}[c=2]\log\left(1-\sigma(h_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}^1,\boldsymbol{y}^2))\right)\right] \tag{4}$$

2. Secondly,

   Remark 3.1. If we obtain a sample $(\boldsymbol{x},\boldsymbol{y}^1,\boldsymbol{y}^2)$ with label 1, it is equivalent to obtain the sample $(\boldsymbol{x},\boldsymbol{y}^2,\boldsymbol{y}^1)$ with label 2.

   Proof.

   $$\begin{aligned}
   \mathsf{Pr}\left(c = 1 \mid \boldsymbol{y}^1,\boldsymbol{y}^2,\boldsymbol{x}\right) &= 1 - \mathsf{Pr}\left(c = 2 \mid \boldsymbol{y}^1,\boldsymbol{y}^2,\boldsymbol{x}\right) \\
   &= 1 - \sigma(s^\natural(\boldsymbol{x},\boldsymbol{y}^2) - s^\natural(\boldsymbol{x},\boldsymbol{y}^1)) \\
   &= 1 - \mathsf{Pr}\left(c = 1 \mid \boldsymbol{y}^2,\boldsymbol{y}^1,\boldsymbol{x}\right) \\
   &= \mathsf{Pr}\left(c = 2 \mid \boldsymbol{y}^2,\boldsymbol{y}^1,\boldsymbol{x}\right)
   \end{aligned}$$

   ■

Well, this information is already encoded in the parameterization of $h_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}^1, \boldsymbol{y}^2)$, i.e., $h_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}^1, \boldsymbol{y}^2) = 1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}^2, \boldsymbol{y}^1)$. We will come back to this point latter since not every method inherently embed this structure in their parameterization.

Because of this, we can rewrite the optimization as:

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \underset{\boldsymbol{x}, \boldsymbol{y}^w, \boldsymbol{y}^l}{\mathbb{E}} \left[-\log \sigma(h_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}^w, \boldsymbol{y}^l)) \right], \tag{5}$$

After solving (5), the alignment fine-tuning is performed by plugging the trained $s_{\boldsymbol{\theta}^\star}$ into (1) and employ any off-the-shelf RL techniques, e.g, PPO [1]. So that is the story of RLHF.

The two-step approach is inherently quite complex and computation intensive. That's where DPO comes to play.

3.2 DPO

An optimal solution of (1) can be derived as follows:

$$\begin{aligned}
&\arg\max_{\pi} \quad \underset{\boldsymbol{x}}{\mathbb{E}} \left[\underset{\boldsymbol{y} \sim \pi(\cdot|\boldsymbol{x})}{\mathbb{E}} \left[s(\boldsymbol{y}, \boldsymbol{x}) - \beta D_{\mathsf{kl}}(\pi \parallel \pi_{\text{ref}}) \right] \right] \\
&= \arg\min_{\pi} \underset{\boldsymbol{x}}{\mathbb{E}} \left[\underset{\boldsymbol{y} \sim \pi(\cdot|\boldsymbol{x})}{\mathbb{E}} \left[\log \frac{\pi(\boldsymbol{y} \mid \boldsymbol{x})}{\exp(\beta^{-1} s(\boldsymbol{y}, \boldsymbol{x})) \pi_{\text{ref}}(\boldsymbol{y} \mid \boldsymbol{x})} \right] \right] \\
&= \arg\min_{\pi} \underset{\boldsymbol{x}}{\mathbb{E}} \left[D_{\mathsf{kl}}(\pi \parallel \frac{1}{Z(\boldsymbol{x})} \exp(\beta^{-1} s(\boldsymbol{y}, \boldsymbol{x})) \pi_{\text{ref}}(\boldsymbol{y} \mid \boldsymbol{x})) \right] \\
&= \frac{1}{Z(\boldsymbol{x})} \exp(\beta^{-1} s(\boldsymbol{y}, \boldsymbol{x})) \pi_{\text{ref}}(\boldsymbol{y} \mid \boldsymbol{x})
\end{aligned}$$

And hence,

$$\pi^\star(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \pi_{\text{ref}}(\boldsymbol{y} \mid \boldsymbol{x}) \exp\left(\beta^{-1} s^\natural(\boldsymbol{y}, \boldsymbol{x}) \right),$$

$$\Leftrightarrow s^\natural(\boldsymbol{y}, \boldsymbol{x}) = \beta \log \frac{\pi^\star(\boldsymbol{y} \mid \boldsymbol{x})}{\pi_{\text{ref}}(\boldsymbol{y} \mid \boldsymbol{x})} + \beta \log Z(\boldsymbol{x})$$

where $\boldsymbol{Z}(\boldsymbol{x})$ is an intractable normalizing factor.

Using the above identity, the generative model in (3) is equivalent to

$$\mathsf{Pr}\left(c = 1 \mid \boldsymbol{y}^1, \boldsymbol{y}^2, \boldsymbol{x}\right) = \sigma\left(\beta \log \frac{\pi^\star(\boldsymbol{y}_1 \mid \boldsymbol{x})}{\pi_{\text{ref}}(\boldsymbol{y}_1 \mid \boldsymbol{x})} - \beta \log \frac{\pi^\star(\boldsymbol{y}_2 \mid \boldsymbol{x})}{\pi_{\text{ref}}(\boldsymbol{y}_2 \mid \boldsymbol{x})} \right) \tag{6}$$

The new generative model is equivalent to (3), while offering a specific parameterization of the score function, i.e,

$$s_{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{x}) = \beta \log \frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x})}{\pi_{\text{ref}}(\boldsymbol{y} \mid \boldsymbol{x})}.$$

Using this parameterization, the optimal solution $\boldsymbol{\theta}^\star$ of the sigmoid criterion (5) directly gives an optimal policy $\pi_{\boldsymbol{\theta}^\star}$ with respect to the fine-tuning objective (1). This solution eliminates the need of RL step as in RLHF.

is it truly equivalent?

3.3 A unified view

Pairwise Preference Generative model.

$$\mathsf{Pr}\left(c = 1 \mid \boldsymbol{y}^1, \boldsymbol{y}^2, \boldsymbol{x}\right) = \sigma\left(q^\natural(\boldsymbol{y}_1, \boldsymbol{x}) - q^\natural(\boldsymbol{y}_2, \boldsymbol{x}) \right),$$

The choice of $q^\natural(\boldsymbol{y}_1, \boldsymbol{x})$ depends on the interest of learning underlying model.

- RLHF wishes to learn the score function: $q^\natural(\boldsymbol{y}_1, \boldsymbol{x}) = s^\natural(\boldsymbol{y}_1, \boldsymbol{x})$
- DPO wishes to learn the optimal policy: $q^\natural(\boldsymbol{y}, \boldsymbol{x}) = \beta \log \dfrac{\pi^\star(\boldsymbol{y} \mid \boldsymbol{x})}{\pi_{\text{ref}}(\boldsymbol{y} \mid \boldsymbol{x})}$

In any case, after deciding the choice for $q^\natural$, we can use maximum likelihood to estimate $q^\natural$ with $q_{\boldsymbol{\theta}}$.

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \underset{\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_\ell}{\mathbb{E}} [-\log \sigma(q_{\boldsymbol{\theta}}(\boldsymbol{y}_w, \boldsymbol{x}) - q_{\boldsymbol{\theta}}(\boldsymbol{y}_\ell, \boldsymbol{x}))] \tag{7}$$

4 IPO

This paper is kind of hard to read as the notation is a bit messed up. I have mixed feeling about it: the idea is quite nice after I figured out all the technical details and fought against my intuition during all the way. However the presentation of their development is quite *cryptic*.

Let's start with their empirical loss:

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad (q_{\boldsymbol{\theta}}(\boldsymbol{y}_w, \boldsymbol{x}) - q_{\boldsymbol{\theta}}(\boldsymbol{y}_\ell, \boldsymbol{x}) - 0.5)^2,$$

where $q_{\boldsymbol{\theta}}$ is defined as in DPO.

This criterion is very counter-intuitive. Specifically,

- Turning a classification-based to a regression-based objective. This is ahhhhhh. But this is just because of my view, nothing wrong here.
- Why regressing toward 0.5? What special about 0.5? And why regressing all possible samples toward 0.5? Does it mean that we treat preference evenly among all samples? While in comparison to (7), the gap $\boldsymbol{q_\theta}(\boldsymbol{y}_w, \boldsymbol{x}) - q_{\boldsymbol{\theta}}(\boldsymbol{y}_l, \boldsymbol{x})$ to be as large as possible.

Well, I have some answered, but not all sadly. So intuition does not help here. Let's dive into the technical development to see if that helps.

4.1 IPO's Loss Derivation

In this section, I attempted to (i) follow and demystify IPO's derivation, make the whole thing a bit more rigorous to identify the key step, and (ii) make a slightly more general formulation.

Note that IPO's goal is to perform the fine-tuning task via (1) as RLHF's and DPO's. What distinguish them is their proposed score function:

$$s^\natural(\boldsymbol{x}, \boldsymbol{y}) = \underset{\boldsymbol{y}' \sim \mu}{\mathbb{E}} \left[v(\boldsymbol{y}, \boldsymbol{y}', \boldsymbol{x}) \right], \tag{8}$$

where the scalar-valued function $v(\boldsymbol{y}, \boldsymbol{y}', \boldsymbol{x}) \in \mathbb{R}$ denotes the degree of preference, i.e., $v(\boldsymbol{y}_1, \boldsymbol{y}_1', \boldsymbol{x}) \succ v(\boldsymbol{y}_2, \boldsymbol{y}_2', \boldsymbol{x})$ implies that the preference of $\boldsymbol{y}_1$ over $\boldsymbol{y}_1'$ is "stronger" compared to the preference comparison of $\boldsymbol{y}_2$ vs $\boldsymbol{y}_2'$. Here, μ *is just some arbitrary policy.*

With that physical meaning, $v(\boldsymbol{y}, \boldsymbol{y}', \boldsymbol{x})$ is any function satisfying:

$$v(\boldsymbol{y}, \boldsymbol{y}', \boldsymbol{x}) \in [\alpha_1, \alpha_2], \ \forall \boldsymbol{y}, \boldsymbol{y}', \boldsymbol{x} \tag{9a}$$

$$v(\boldsymbol{y}, \boldsymbol{y}, \boldsymbol{x}) = 0.5(\alpha_2 + \alpha_1), \ \forall \boldsymbol{y}, \boldsymbol{x} \tag{9b}$$

$$v(\boldsymbol{y}, \boldsymbol{y}', \boldsymbol{x}) = \alpha_1 + \alpha_2 - v(\boldsymbol{y}', \boldsymbol{y}, \boldsymbol{x}) \tag{9c}$$

Condition (9a) is to avoid DPO's issue in which over-fitting leads to policy deviating arbitrarily far away from π regardless of β. Condition (9b) and (9c) is to enforce the physical meaning of pairwise preference. With that score function, IPO's generative model can be seen as follows.

IPO's Pairwise Generative Model. The binary label $c \in \{1, 2\}$ for sample $\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{x}$ is

$$\mathsf{Pr}\,(c = 1) \propto v(\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{x}) \tag{10}$$

Here, binary RV $c = 1$ indicates $\boldsymbol{y}_1 \succ \boldsymbol{y}_2$. The likelihood of this even happening is proportional to $v(\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{x})$, while the exact probability is unknown. From modeling perspective, the lack of specification in (10) relative to the BT model makes it more general.

Similar to DPO, the optimal policy to IPO's objective in (1) is

$$\pi^\star(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \pi_{\text{ref}}(\boldsymbol{y} \mid \boldsymbol{x}) \exp\left(\beta^{-1} \mathop{\mathbb{E}}_{\boldsymbol{y}' \sim \mu} \left[v(\boldsymbol{y}, \boldsymbol{y}', \boldsymbol{x})\right]\right),$$

which enables the trick originally made by DPO by comparing 2 responses:

$$q^\star(\boldsymbol{y}_1, \boldsymbol{x}) - q^\star(\boldsymbol{y}_2, \boldsymbol{x}) = \mathop{\mathbb{E}}_{\boldsymbol{y} \sim \mu} \left[v(\boldsymbol{y}_1, \boldsymbol{y}, \boldsymbol{x}) - v(\boldsymbol{y}_2, \boldsymbol{y}, \boldsymbol{x})\right], \tag{11}$$

where $q^\star(\boldsymbol{y}, \boldsymbol{x}) = \beta \log \pi^\star(\boldsymbol{y} \mid \boldsymbol{x}) - \beta \log \pi_{\text{ref}}(\boldsymbol{y} \mid \boldsymbol{x})$. Let $q_{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{x}) = \beta \log \pi_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) - \beta \log \pi_{\text{ref}}(\boldsymbol{y} \mid \boldsymbol{x})$. All we want is to enforce the equality (11) for all $\boldsymbol{y}_1, \boldsymbol{y}_2$ with respect to $q_{\boldsymbol{\theta}}$, and one way to realize it is using squared loss

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \mathop{\mathbb{E}}_{\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2 \sim \mu} \left[\left(q_{\boldsymbol{\theta}}(\boldsymbol{y}_1, \boldsymbol{x}) - q_{\boldsymbol{\theta}}(\boldsymbol{y}_2, \boldsymbol{x}) - \mathop{\mathbb{E}}_{\boldsymbol{y} \sim \mu} \left[v(\boldsymbol{y}_1, \boldsymbol{y}, \boldsymbol{x}) - v(\boldsymbol{y}_2, \boldsymbol{y}, \boldsymbol{x})\right] \right)^2 \right]$$

Expanding the squared term, the optimization problem reduces to

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}, \\ \boldsymbol{y}_1, \boldsymbol{y}_2 \sim \mu}} \left[(q_{\boldsymbol{\theta}}(\boldsymbol{y}_1, \boldsymbol{x}) - q_{\boldsymbol{\theta}}(\boldsymbol{y}_2, \boldsymbol{x}))^2 - 2[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1, \boldsymbol{x}) - q_{\boldsymbol{\theta}}(\boldsymbol{y}_2, \boldsymbol{x})] \mathop{\mathbb{E}}_{\boldsymbol{y} \sim \mu} \left[v(\boldsymbol{y}_1, \boldsymbol{y}, \boldsymbol{x}) - v(\boldsymbol{y}_2, \boldsymbol{y}, \boldsymbol{x})\right] \right] \tag{12}$$

The issue lays in the unknown factor in the second term. Now comes to the IPO's peculiar derivations. The second term can be written as

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\boldsymbol{y}_1, \boldsymbol{y}_2 \sim \mu} \left[[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1, \boldsymbol{x}) - q_{\boldsymbol{\theta}}(\boldsymbol{y}_2, \boldsymbol{x})] \mathop{\mathbb{E}}_{\boldsymbol{y} \sim \mu} \left[v(\boldsymbol{y}_1, \boldsymbol{y}, \boldsymbol{x}) - v(\boldsymbol{y}_2, \boldsymbol{y}, \boldsymbol{x})\right] \right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y} \sim \mu} \left[\left(q_{\boldsymbol{\theta}}(\boldsymbol{y}_1, \boldsymbol{x}) - q_{\boldsymbol{\theta}}(\boldsymbol{y}_2, \boldsymbol{x})\right) \left(v(\boldsymbol{y}_1, \boldsymbol{y}, \boldsymbol{x}) - v(\boldsymbol{y}_2, \boldsymbol{y}, \boldsymbol{x})\right) \right] \\
&= \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y} \sim \mu} \left[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1, \boldsymbol{x}) v(\boldsymbol{y}_1, \boldsymbol{y}, \boldsymbol{x}) + q_{\boldsymbol{\theta}}(\boldsymbol{y}_2, \boldsymbol{x}) v(\boldsymbol{y}_2, \boldsymbol{y}, \boldsymbol{x}) \right]}_{(*)} \\
&\qquad - \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y} \sim \mu} \left[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1, \boldsymbol{x}) v(\boldsymbol{y}_2, \boldsymbol{y}, \boldsymbol{x}) + q_{\boldsymbol{\theta}}(\boldsymbol{y}_2, \boldsymbol{x}) v(\boldsymbol{y}_1, \boldsymbol{y}, \boldsymbol{x}) \right]}_{(**)}
\end{aligned}$$

For (*), notice that the two terms are essentially the same under expectation when $\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}$ are i.i.d.

$$\begin{aligned}
(*) &= \mathop{\mathbb{E}}_{\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}} \left[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1, \boldsymbol{x}) v(\boldsymbol{y}_1, \boldsymbol{y}, \boldsymbol{x}) + q_{\boldsymbol{\theta}}(\boldsymbol{y}_2, \boldsymbol{x}) v(\boldsymbol{y}_2, \boldsymbol{y}, \boldsymbol{x}) \right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{y}_1, \boldsymbol{y}} \left[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1, \boldsymbol{x}) v(\boldsymbol{y}_1, \boldsymbol{y}, \boldsymbol{x}) \right] + \mathop{\mathbb{E}}_{\boldsymbol{y}_2, \boldsymbol{y}} \left[q_{\boldsymbol{\theta}}(\boldsymbol{y}_2, \boldsymbol{x}) v(\boldsymbol{y}_2, \boldsymbol{y}, \boldsymbol{x}) \right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{y}_1, \boldsymbol{y}_2} \left[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1, \boldsymbol{x}) v(\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{x}) \right] + \mathop{\mathbb{E}}_{\boldsymbol{y}_1, \boldsymbol{y}} \left[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1, \boldsymbol{x}) v(\boldsymbol{y}_1, \boldsymbol{y}, \boldsymbol{x}) \right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{y}_1, \boldsymbol{y}_2} \left[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1, \boldsymbol{x}) v(\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{x}) \right] + \mathop{\mathbb{E}}_{\boldsymbol{y}_1, \boldsymbol{y}_2} \left[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1, \boldsymbol{x}) v(\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{x}) \right] \\
&= 2 \mathop{\mathbb{E}}_{\boldsymbol{y}_1, \boldsymbol{y}_2} \left[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1, \boldsymbol{x}) v(\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{x}) \right]
\end{aligned} \tag{13}$$

This trick is pretty cool ha ;), I also used this one in one of my proof :))) [x]. Note that the particular name of the variables under expectation might not matter as long as they share the same distribution. For example,

$$\mathop{\mathbb{E}}_{x \sim p} \left[f(x)\right] + \mathop{\mathbb{E}}_{y \sim p} \left[f(y)\right] = \mathop{\mathbb{E}}_{x \sim p} \left[f(x)\right] + \mathop{\mathbb{E}}_{x \sim p} \left[f(x)\right] = 2 \mathop{\mathbb{E}}_{x \sim p} \left[f(x)\right].$$

Similar trick can be applied for $(**)$ as well:

$$\begin{aligned}(**) &= \mathop{\mathbb{E}}_{\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{y}\sim\mu}[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1,\boldsymbol{x})v(\boldsymbol{y}_2,\boldsymbol{y},\boldsymbol{x}) + q_{\boldsymbol{\theta}}(\boldsymbol{y}_2,\boldsymbol{x})v(\boldsymbol{y}_1,\boldsymbol{y},\boldsymbol{x})] \\ &= 2\mathop{\mathbb{E}}_{\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{y}\sim\mu}[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1,\boldsymbol{x})v(\boldsymbol{y}_2,\boldsymbol{y},\boldsymbol{x})] \\ &= 2\mathop{\mathbb{E}}_{\boldsymbol{y}_1\sim\mu}[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1,\boldsymbol{x})]\mathop{\mathbb{E}}_{\boldsymbol{y}_2,\boldsymbol{y}}[v(\boldsymbol{y}_2,\boldsymbol{y},\boldsymbol{x})] \\ &= (\alpha_1+\alpha_2)\mathop{\mathbb{E}}_{\boldsymbol{y}_1\sim\mu}[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1,\boldsymbol{x})] \end{aligned} \tag{14}$$

where the last equality holds because $\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}$ are independent and $\mathbb{E}_{\boldsymbol{y}_2,\boldsymbol{y}\sim\mu}[v(\boldsymbol{y}_2,\boldsymbol{y},\boldsymbol{x})] = 0.5(\alpha_1+\alpha_2)$, which in turn can be derived from conditions (9b) and (9c). Combining (13), (14) gives

$$\begin{aligned}(*)-(**) &= 2\mathop{\mathbb{E}}_{\boldsymbol{y}_1,\boldsymbol{y}_2}[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1,\boldsymbol{x})v(\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{x})] - (\alpha_1+\alpha_2)\mathop{\mathbb{E}}_{\boldsymbol{y}_1}[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1,\boldsymbol{x})] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{y}_1,\boldsymbol{y}_2}[2q_{\boldsymbol{\theta}}(\boldsymbol{y}_1,\boldsymbol{x})v(\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{x}) - (v(\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{x})+v(\boldsymbol{y}_2,\boldsymbol{y}_1,\boldsymbol{x}))q_{\boldsymbol{\theta}}(\boldsymbol{y}_1,\boldsymbol{x})] \quad \text{(by (9c))} \\ &= \mathop{\mathbb{E}}_{\boldsymbol{y}_1,\boldsymbol{y}_2}[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1,\boldsymbol{x})v(\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{x}) - v(\boldsymbol{y}_2,\boldsymbol{y}_1,\boldsymbol{x})q_{\boldsymbol{\theta}}(\boldsymbol{y}_1,\boldsymbol{x})] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{y}_1,\boldsymbol{y}_2}[q_{\boldsymbol{\theta}}(\boldsymbol{y}_1,\boldsymbol{x})v(\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{x}) - v(\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{x})q_{\boldsymbol{\theta}}(\boldsymbol{y}_2,\boldsymbol{x})] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{y}_1,\boldsymbol{y}_2}[(q_{\boldsymbol{\theta}}(\boldsymbol{y}_1,\boldsymbol{x})-q_{\boldsymbol{\theta}}(\boldsymbol{y}_2,\boldsymbol{x}))v(\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{x})] \end{aligned}$$

The expression in (12) becomes

$$\begin{aligned}&\arg\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x},\\ \boldsymbol{y}_1,\boldsymbol{y}_2\sim\mu}}\left[(q_{\boldsymbol{\theta}}(\boldsymbol{y}_1,\boldsymbol{x})-q_{\boldsymbol{\theta}}(\boldsymbol{y}_2,\boldsymbol{x}))^2\right] - 2\mathop{\mathbb{E}}_{\substack{\boldsymbol{x},\\ \boldsymbol{y}_1,\boldsymbol{y}_2\sim\mu}}[(q_{\boldsymbol{\theta}}(\boldsymbol{y}_1,\boldsymbol{x})-q_{\boldsymbol{\theta}}(\boldsymbol{y}_2,\boldsymbol{x}))v(\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{x})] \\ &= \arg\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x},\\ \boldsymbol{y}_1,\boldsymbol{y}_2\sim\mu}}\left[(q_{\boldsymbol{\theta}}(\boldsymbol{y}_1,\boldsymbol{x})-q_{\boldsymbol{\theta}}(\boldsymbol{y}_2,\boldsymbol{x})-v(\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{x}))^2\right] \\ &\approx \arg\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x},\\ \boldsymbol{y}_1,\boldsymbol{y}_2\sim\mu}}\left(q_{\boldsymbol{\theta}}(\boldsymbol{y}_1,\boldsymbol{x})-q_{\boldsymbol{\theta}}(\boldsymbol{y}_2,\boldsymbol{x})-\mathbb{I}[c=1]\alpha_2-\mathbb{I}[c=2]\alpha_1\right)^2, \end{aligned} \tag{15}$$

where $c \in \{1,2\}$ is the pairwise preference label. The approximation is very crude: using $\{\alpha_1,\alpha_2\}$ to approximate $v(\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{x})$. In a more general view, this is a non-parametric estimation of $v(\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{x})$ under the generative model (10) using only 1 sample c. Note that this estimation is made *independently* to the alignment learning, i.e, optimizing (15). It is also the reason why IPO claims that they **even don't need to learn the score function**.

A further variance-reduced improvement made in IPO is to consider the symmetrical property of the sample: if we obtain the triple $(\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{x})$ with label c, then it is legitimate to assume another triple $(\boldsymbol{y}_2,\boldsymbol{y}_1,\boldsymbol{x})$ with label $2/c$. With that in mind, we arrive at the final criterion:

$$\begin{aligned}&\arg\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x},\\ \boldsymbol{y}_1,\boldsymbol{y}_2\sim\mu}}\left[(h_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_1,\boldsymbol{y}_2)-\mathbb{I}[c=1]\alpha_2-\mathbb{I}[c=2]\alpha_1)^2+(h_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_2,\boldsymbol{y}_1)-\mathbb{I}[c=1]\alpha_1-\mathbb{I}[c=2]\alpha_2)^2\right] \\ &= \arg\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x},\\ \boldsymbol{y}_1,\boldsymbol{y}_2\sim\mu}}\left[(h_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_\ell)-\alpha_2)^2+(h_{\boldsymbol{\theta}}(\boldsymbol{x}_i,\boldsymbol{y}_{i,l},\boldsymbol{y}_{i,w})-\alpha_1)^2\right] \\ &= \arg\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x},\\ \boldsymbol{y}_1,\boldsymbol{y}_2\sim\mu}}\left[(h_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_\ell)-0.5(\alpha_1+\alpha_2))^2\right] \end{aligned} \tag{16}$$

IPO's particular choice. In IPO, they settle with $v(\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{x}) = \mathsf{Pr}(\boldsymbol{y}_1 \succ \boldsymbol{y}_2 \mid \boldsymbol{x})$, and hence $\alpha_1 = 0, \alpha_2 = 1$. While it is looking sensible, any other arbitrary choice should be as valid. Anyhow, it leads to their criterion

$$\operatorname*{minimize}_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x},\\ \boldsymbol{y}_1,\boldsymbol{y}_2\sim\mu}}\left[(h_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_\ell)-0.5)^2\right]$$

Effect of α_1,α_2. Firstly, the final criterion (16) reveals that only the sum $\alpha_1+\alpha_2$ matters, not the absolute values of α_1,α_2. Large $\alpha_1+\alpha_2$ means a broader preference model. However, it also leads to higher estimation error.

4.2 IPO's loss discussion

- There is no score learning. The chosen score function has no parameter to estimate. This is contrast to RLHF/DPO's approach where we known functional form of the score function governed by parameter $\boldsymbol{\theta}$.
- The seemingly disconnection between the score function and the preference model. However, in reality, they are connected via: μ and the oracle preference probability $\mathsf{Pr}\left(\boldsymbol{y}_1 \succ \boldsymbol{y}_2 \mid \boldsymbol{x}\right) = f^\natural(\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{x})$, where $f^\natural$ is any function satisfying $f^\natural(\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{x}) \in [0, 1]$, $f^\natural(\boldsymbol{y}, \boldsymbol{y}, \boldsymbol{x}) = 0.5$, and $f^\natural(\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{x}) + f^\natural(\boldsymbol{y}_2, \boldsymbol{y}_1, \boldsymbol{x}) = 1$. Okay, we are getting some structure here. Under this view, the preference data directly dictates the score function.
- There is no learning, there is no overfitting. We only use approximation
- In terms of preference model, anything is feasible, including contradicting preferences, i.e, $\boldsymbol{y}_1 \succ \boldsymbol{y}_2, \boldsymbol{y}_2 \succ \boldsymbol{y}_3, \boldsymbol{y}_3 \succ \boldsymbol{y}_1$. The BT model does not allow this. Therefore, if this happen because of noisy labels, IPO would just adapt to it while RLHF/DPO won't.
- Although the preference model is very liberal, i.e, allow for contradicting preferences, the objective implies that there is a total ordering over responses $\boldsymbol{y}$.
- The score depends on μ, which is not necessarily a good feature.
- There could be this situation: $\mathsf{Pr}\left(\boldsymbol{y}_1 \succ \boldsymbol{y}_2\right)$ and $s(\boldsymbol{y}_1, \boldsymbol{x}) < s(\boldsymbol{y}_2, \boldsymbol{x})$.
- But the $f_{\boldsymbol{\theta}}(\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{x}) = q_{\boldsymbol{\theta}}(\boldsymbol{y}_1, \boldsymbol{x}) - q_{\boldsymbol{\theta}}(\boldsymbol{y}_2, \boldsymbol{x})$ has structure.
- Their motivation is to address DPO's weakness. To me, it is more like a by-produce that they archive that goal. Let's see.

5 Experiment Design

We want to show a single point: IPO's approximation could be terribly wrong, and hence, it could not reach the optimal solution. In the meantime, DPO attain optimal solution easily. But can we design generative models such that the preference labels for both IPO and DPO are the same? The problem is the score functions are not the same.

Let assume bandit, 3 actions y_0, y_1, y_2. For DPO, assume

$$s^\natural_{\text{DPO}}(\boldsymbol{y}) = [10.000, 9.593, 7.800] \tag{17}$$

$$s^\natural_{\text{IPO}}(y) = \mathop{\mathbb{E}}_{y'}[\mathsf{Pr}\left(y \succ y'\right)] \tag{18}$$

There are 2 factors: score function, and preference generative model.

Method	Score function	Preference generative model
DPO	$s^\natural(\boldsymbol{y})$	$\mathsf{Pr}\left(c = 1\right) = \sigma(s^\natural(\boldsymbol{y}_1) - s^\natural(\boldsymbol{y}_2))$
IPO	$\mathbb{E}_{\boldsymbol{y}' \sim \mu}[v^\natural(\boldsymbol{y}, \boldsymbol{y}')]$	$\mathsf{Pr}\left(c = 1\right) = v^\natural(\boldsymbol{y}_1, \boldsymbol{y}_2)$

Score function of IPO has structure, and it implies this interesting property: total score is a constant. Not sure how to feel about it!